Many computers and microcontrollers are programmed in hexadecimal, which represents numerical values using the numerals 0 through 9 plus the letters A through F. Entering the numerals is quite easy with the numeric keypad of a PC keyboard, but the letters are scattered over the entire alphanumeric portion of the keyboard. This article describes the construction of a keypad with 20 buttons, which includes all the necessary characters as well as several control characters.

Design by U. Reiser

hexadecimal keypad for PCs

problem–free programming

The hexadecimal keypad is connected between the actual PC and its keyboard. It also provides the necessary extension for the keyboard cable. The following functions are provided by this keypad:

- generating the lower–case letters a through f;
- generating the upper–case letters A through F when the Shift or Caps Lock key of the PC keyboard is activated. Activating the # key on the hex keypad will prevent characters from being output by the numeric/shifted character keys located in the alphanumeric portion of the PC keyboard);
- generating the numerals 0 through 9, corresponding to the numeric keys of the alphanumeric portion of the PC keyboard;
- generating the numerals 0 through 9, corresponding to the keys of the PC keyboard numeric keypad, when the # key of the hex keypad and the NUM key of the PC keyboard are activated;
- generating cursor control characters corresponding to the keys of the PC numeric keypad, if the # key of the

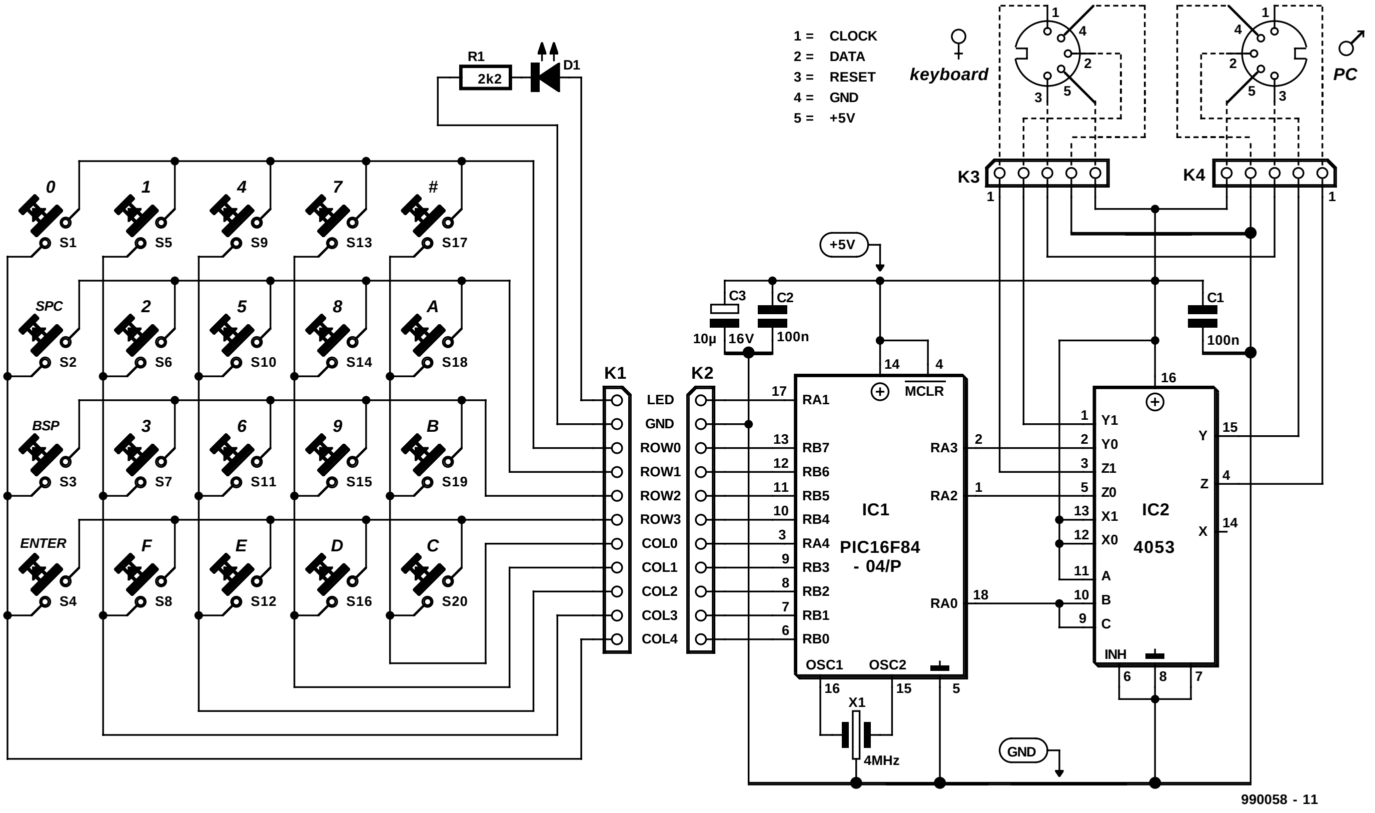


Figure 1. Circuit diagram of the hexadecimal keypad.

hex keypad is activated and the NUM key of the PC keyboard is deactivated;

- generating the Space, Backspace and Enter characters (in all modes).

- In addition, it is possible to utilize the control functions of the numeric keypad. This requires that the NUM key of the PC keyboard is not active, which is indicated by the NUM LOCK LED of the keyboard not being illuminated.

The schematic diagram of the hex keypad, as shown in **Figure 1** shows that only a few components are needed in addition to the pushbutton switches. The operation of the hardware is easily explained. The PIC controller (16F84) constantly polls the 20 switches and sends the code corresponding to an activated switch key to the PC. The second IC, a 4053, is a bidirectional multiplexer that connects either the PC keyboard or the hex keypad to the PC. The keypad is connected to the PC only when its controller has detected a keypress, and the multiplexer immediately reconnects the keyboard to the PC after the key code has been transmitted by the hex keypad. It is at least theoretically possible for the hex keypad to interrupt signals being sent from the PC keyboard. The PC supplies the 5 V operating voltage for the hex keypad (and also for the PC keyboard).

Software

You can see how the program works from **Figure 2**, as well as from the assembly-language listing on the *Elektor Electronics* Readers Services diskette (order code **996021-1**). In the first section of the program, labelled Initialisation, the input and output functions of ports A and B are set up and the internal pull–up resistors for port B are enabled. The subsequent power–on self test flashes the LED of the hex keypad twice. The on and off intervals are both set to 109 ms by TIME4. This function is intended to indicate to the user that the basic PIC processor is operational.

The main program of the controller starts with polling the key matrix (S1 through S20). This is performed by setting one of the five columns from High to Low (COL??) and waiting for 5 ms (BOUNCE), after which all four rows are polled for the presence of a Low level passed through by any activated switch. Based on the arrangement of the switches in rows and columns, the switch number is selected and the program branches to the appropriate subroutine (SW??). There it waits 5 ms, using

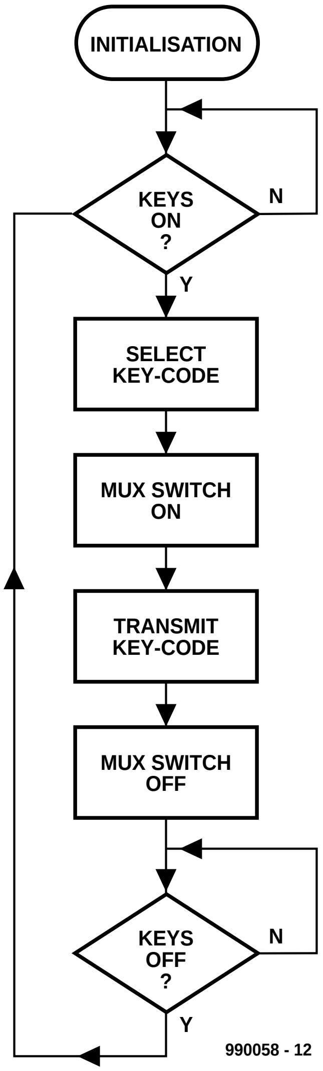


Figure 2. Software flow chart.

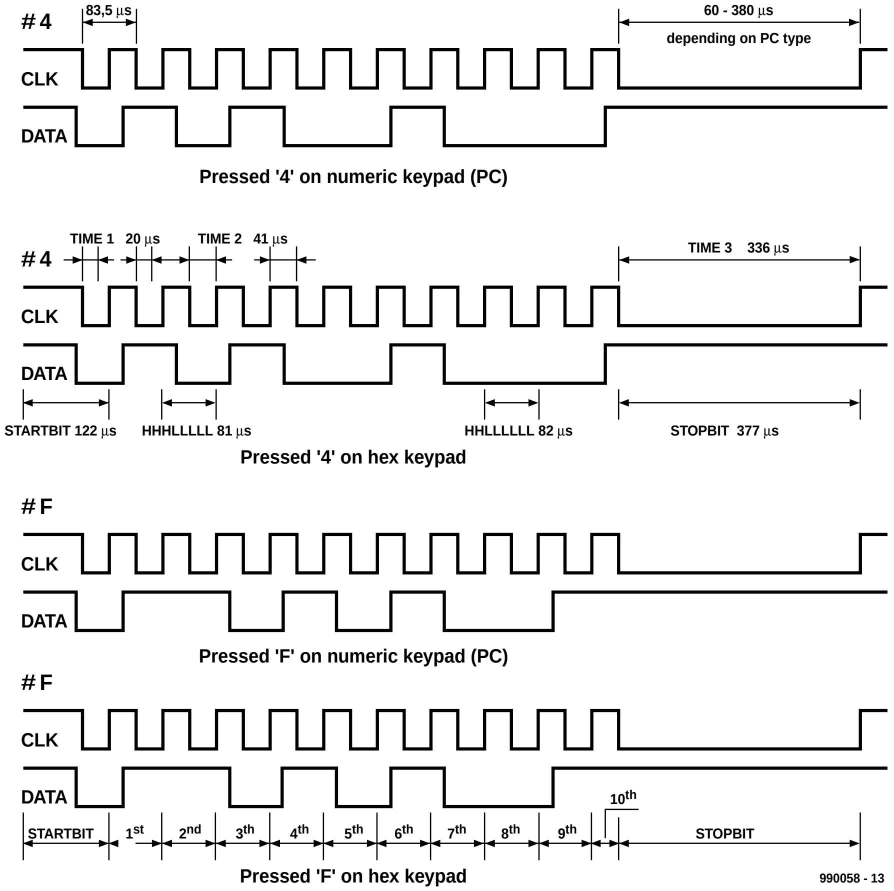
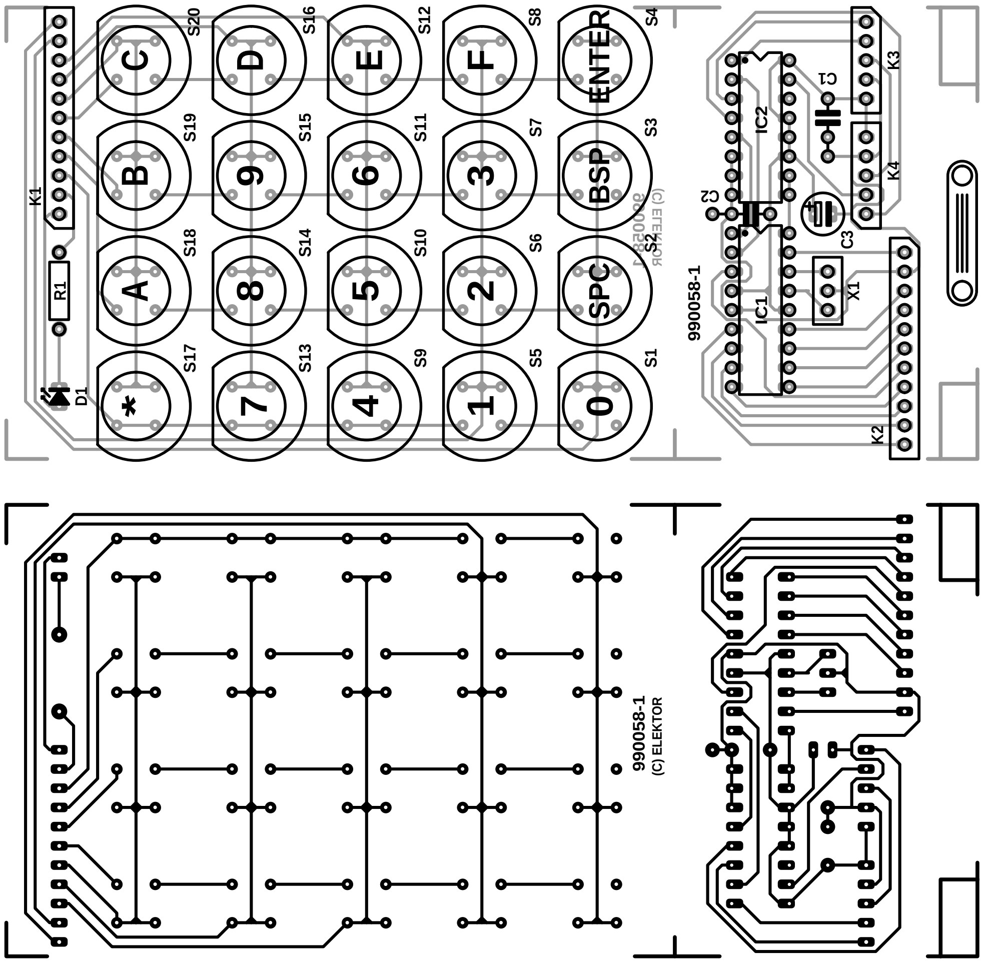


Figure 3. Key code pulse sequences.

CALL BOUNCE, and then again polls the already detected switch. If the result is negative, an error is assumed (due to keybounce, for example). GOTO COLUMN? repeats the polling of the same column. If on the other hand the second polling confirms the first result, the LED is polled (in the case of the numeric keys) and CALL RELAISON (call multiplexer-on) sets the B and C control inputs of the 4053 to Low, which causes the data and clock signals of the hex keypad to be connected to the PC. In order to assure reliable switching of IC2, program execution is delayed for an additional 5 ms by again calling the timer loop BOUNCE.

The bit sequence for the key code is generated by calling twelve subroutines. Following this, the instruction BSF PORTA,REL causes the 4053 multiplexer to be switched back to its prior state. To avoid repeated key code generation for a pressed key, the routine GOTO KEYSOFF waits for all keys to be released. This is checked by setting all columns to Low and polling all four rows in turn. Searching for a new key press is

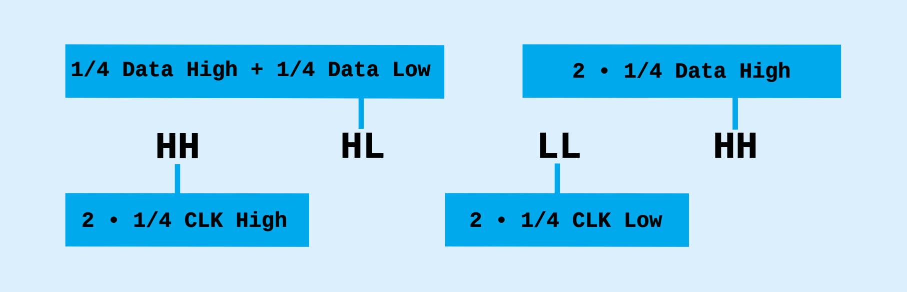
Figure 4. Printed circuit board layout and component mounting diagram.



initiated by the instruction GOTO COLUMN4 only after no Low level is detected.

Key codes

Figure 3 shows two pulse sequences that represent the signals on the clock and data lines for the keys 4 and F. The clock frequency is approximately 12 kHz. Level changes on the data line, when required, occur halfway through the High interval of the clock signal. Since the beginning and end of a pulse train are always the same, their generation is relegated to the separate subroutines STARTBIT and STOPBIT. The data and clock signals are generated by setting and clearing RA2 and RA3 as appropriate. The pulse length is determined by the two timing loops TIME1 and TIME2. If a change in the level of the data signal is required (HHHL and HHLH), TIME1 generates an interval of 20 μs, which is a quarter of the clock interval. After this, the level changes, and a second quarter–clock interval completes the High portion of the clock signal. If no level change occurs (HHLL and HHHH), the High portion of the signal (half–clock interval) is completely produced by TIME2 (41 µs). Since a level change never occurs during the Low portion of the clock signal, this portion is always generated by TIME2. Altogether, this procedure results in a pulse period of 81 to 82 ms, which corresponds to a frequency of 12.345 to 12.195 kHz. Since the pulse sequences for the individual keys are made up of several, partly identical subsequences, these are implemented as subroutines (HHLLLLLL, HHHLLLHH and so on). The meaning of the arrangement of the letters H and L in the names of the subroutines is as follows:



COMPONENTS LIST

Resistor:
R1 = 390Ω

Capacitor:
C1 = 100nF

Semiconductors:
IC1 = PIC16F84 (ready-programmed, order code 996516-1)
IC2 = 4053
D1 = LED, red

Miscellaneous:
S1-S20 = presskey type D6-R-RD (Conrad Electronics # 70 84 88)
K1,K2 = 22-way pinheader
K3,K4 = 10-way SIL pinheader
5-way DIN plug (or mini-DIN) for chassis mounting
5-way DIN socket (or mini-DIN) for chassis mounting
Enclosure type Pult-box 25 x, 109x70x50mm (Conrad Electronics # 52 24 49)
Strain relief grommet (Conrad Electronics # 52 46 20)
2 PCB standoffs, M3x10mm
Flatcable, 11-way, length approx. 10 cm
PCB, order code **990058-1**
Disk, source code file, order code **996021-1**.

Hardware

The dimensions of the circuit boards (see **Figure 4** and the layout of the components) are designed to fit the enclosure recommended in the list of components. To best utilize the available space, the keypad portion (with the LED and series resistor) and the processor portion (with the PIC and the multiplexer) are mounted on separate printed circuit boards. These are connected by an 11-lead flat cable via connectors K1 and K2. You should use the switches recommended in the list of components, since they internally connect certain circuit board tracks. If other types of switches are used, it may be necessary to fit wire jumpers so that every switch connects two sets of contacts when it is activated, as is already the case with S9, S13 and S17.

The two circuit boards are fitted one on top of the other using 10-mm standoffs. The connections to the PC and the keyboard are provided by K3/K4 and a DIN or mini-DIN plug and jack set. The two 5-lead cables are attached to the processor circuit board by strain–relief clamps.

Make openings for the 20 switches and the LED in the enclosure, and then fit the circuit-board sandwich into the enclosure. Now cut two strips of circuit board material (without copper overlay) 65 mm long and 9 to 10 mm wide (depending on the type of switch used). Fit one of these into the guide slots of the enclosure at the level of switches S1 through S4, and the other one over the processor circuit board (both crossways). These prevent the switches from being pushed down into the enclosure when they are pressed. On the front side of the enclosure, cut a rectangular hole to allow the two cables to pass to the outside.

Assuming that all the components have been properly fitted, it is not necessary to specifically test the operation of the keypad. Since the keypad draws its power from the PC, you should try it out with a different power source before putting it into service, just to be safe. The LED should flash twice after the 5 V supply voltage is connected, to indicate that the PIC is operational. Switching the LED on and off with the # key can also be used as a simple functional test. If a switch is permanently activated, possibly due to its being jammed by the enclosure or incorrectly fitted, the circuit will not be able to do anything except to flash the LED twice! You can use an oscilloscope or logic probe to check the signals on the data and control outputs. When a key is pressed, pulse sequences similar to those already shown should be seen on the clock and data outputs, and a single pulse on the control output (RA1).

(990058–1)